Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №5

по курсу «Организация ЭВМ и систем»

**Вариант 9**

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Клюкин В.Л./

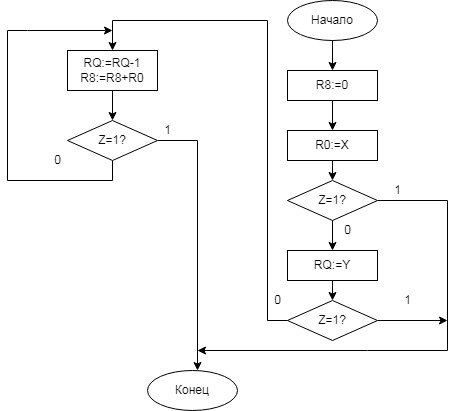
Киров 2022

1. Задание

Выполнить умножение положительных чисел с фиксированной запятой методом "младшими разрядами вперед". Произведение представлено 16-разрядным, а множители 8-разрядными двоичными числами

2. Определение архитектуры и программирование

2.1. Схема алгоритма



2.2. Форматы данных

По заданию множители представлены 8-разрядными числами, произведение – 16-разрядными числами.

2.3. Программно-доступные регистры

ЭВМ имеет девять программно-доступных регистров: шесть регистров общего назначения (r0-r5), программный счетчик – IP (r6), регистр признаков – FLAGS (r7), содержащий разряд признака нуля (Z), а также регистр указателя стека – SP (r8).

2.4. Система команд

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Мнемоника | Описание | Изменение признака Z |
| Суммирование | ADD r r\* | r:=r+r\*, IP=IP+1 | + |
| Вычитание | SUB r r\* | r:=r-r\*, IP=IP+1 | + |
| Добавление С | AD r C | r:=r+C, IP=IP+1 | + |
| Вычитание С | SB r C | r:=r-C, IP=IP+1 | + |
| Чтение в регистр | LD r A | r:=M[A], IP=IP+1 | - |
| Запись регистра | MV r A | M[a]:=r, IP=IP+1 | - |
| Чтение в регистр с индексацией | LDI r r\* | r:=M[r\*], IP=IP+1 | - |
| Запись в стек | PUSH r (SP) | M[SP]:=r, SP:=SP-1; IP:=IP+1 | - |
| Чтение из стека | POP r (SP) | SP:=SP+1, r:=M[SP], IP:=IP+1 | - |
| Переход | JMP A | IP:=A | - |
| Переход, если нуль | JZ A | Если Z=1, то IP:=A  иначе IP:=IP+1 | - |
| Обращение к подпрограмме | CALL (SP) A | M[SP]:=IP, SP:=SP-1 IP:=IP+1 | - |
| Возврат из подпрограммы | RET (SP) | SP:=SP+1, IP:=M[SP] | - |
| Остановка | HLT A | IP:=A, остановка | - |

В описании системы команд приняты следующие обозначения: ⎯ r, r\*∈ {r0, r1, … r8} – программно-доступные регистры: регистр r\* является источником данных, а регистр r – приемником результата, но может также служить источником второго операнда ⎯ М[А] – ячейка памяти с адресом А ⎯ Знак "+" в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак "-" свидетельствует о сохранении старого значения признака

2.5. Программа

AASP – адрес ячейки памяти, в которой находится адрес начала накопителя стека;

AAM – адрес ячейки памяти, в которой находится начальный адрес массива исходных данных (Y1,X1,Y2,X2,…,YN,XN);

AD – начальный адрес подпрограммы деления чисел нацело;

AS – адрес ячейки, в которую записывается сумма S;

AP – адрес ячейки памяти, в которую помещается значение признака переполнения;

SA – начальный адрес программы суммирования частных.

LD r8 AASP Загрузка регистра указателя стека SP

(начальная установка)

LD r5 AAM Загрузка адреса массива AM в регистр r5

LD r4 AN Загрузка числа повторений цикла N в регистр r4

SUB r3 r3 Очистка регистра r3 для суммы S

SUB r2 r2 Очистка регистра r2 для признака Q

m3 LDI r1 (r5)+ Чтение делителя Y в регистр r1

BEQ m1 Если PZ=1 (Y=0), то переход на метку m1

LDI r0 (r5)+ Чтение делимого X в регистр r0

CALL AD Обращение к подпрограмме по адресу AD

ADD r3 r1 Суммирование

SB r4 "1" Вычитание единицы из числа повторений цикла

BEQ m2 Если N=0, то переход на метку m2

BR m3 Переход на метку m3

m1 SB r2 "1" Запись единиц в регистр признака Q

m2 MV r3 AS Запись суммы S адресу AS

MV r2 AQ Запись признака Q адресу AQ

HLT SA Загрузка PC и останов

2.6. Распределение программно-доступных регистров ЭВМ

|  |  |  |
| --- | --- | --- |
|  | Регистр ЭВМ |  |
| r0 |  |  |
| r1 | Xi | Множитель 1-ый |
| r2 | Yi | Множитель 2-ой |
| r3 | S | Произведение |
| r4 | N | Число повторений цикла |
| r5 | ARRAY | Указатель массива |
| r6 | IP | Программный счетчик |
| r7 | FLAGS|Z | Регистр признаков |
| r8 | SP | Регистр указателя стека |

AD R2 00 Проверка Ni на равенство нулю

JZ m3 Если Ni = 0, то переход на метку m3

m4: ADD R3 R1 R3:=R3+R1

JZ m3 Если r1 = 0, то переход на метку m3

SB R2 01 Уменьшение кол-ва сдвигов Ni на 1

JZ m3 Если Ni = 0, то переход на метку m3

JMP m4 Переход на метку m4

m3: RET Возврат из подпрограммы

3. Кодирование программы и распределение памяти программ и данных

3.1. Форматы данных

Команды ЭВМ имеют четыре формата и в зависимости от признака формата (Ф) и кода операции (К) делятся на четыре группы (рис. 6.6). Ф

3.2. Коды операций

|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Суммирование | ADD r r\* | 01 |
| Вычитание | SUB r r\* | 02 |
| Добавление С | AD r C | 9 |
| Вычитание С | SB r C | A |
| Чтение в регистр | LD r A | B |
| Запись регистра | MV r A | C |
| Чтение в регистр с индексацией | LDI r r\* | 0E |
| Запись в стек | PUSH r (SP) | 07 |
| Чтение из стека | POP r (SP) | 06 |
| Переход | JMP A | 03 |
| Переход, если нуль | JZ A | 04 |
| Обращение к подпрограмме | CALL (SP) A | D |
| Возврат из подпрограммы | RET (SP) | 05 |
| Остановка | HLT A | 00 |

3.3. Распределение памяти программ и данных

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 00 | 0010 | IP | Начало программы |
| 01 | 0040 | BP | Указатель стека |
| 02 | 0006 | ARRAY | Указатель на массив |
| 03 | 0004 | N | Число повторений |
| 04 |  | SUM | Сумма |
| 05 |  | - |  |
| 06 |  | X0 | Массив |
| 07 |  | Y0 |  |
| 08 |  | X1 |  |
| 09 |  | Y1 |  |
| 0A |  | X2 |  |
| 0B |  | Y2 |  |
| 0C |  | X3 |  |
| 0D |  | Y3 |  |
| 0E |  |  |  |
| 0F |  |  |  |
| 10 | B801 | LD SP BP | Загрузка регистра указателя стека |
| 11 | B502 | LD R5 ARRAY (M1) | Загрузка адреса массива в регистр 5 |
| 12 | B403 | LD R4 N | Загрузка кол-ва посвторений в 4 регситр |
| 13 | 0233 | SUB R3 R3 | Очистка регистра 3 |
| 14 | 0E15 | LDI R1 [R5] | Загрузка |
| 15 | 9501 | AD R5 01 | +1 |
| 16 | 0E25 | LDI R2 [R5] | Загрузка |
| 17 | 9501 | AD R5 01 | +1 |
| 18 | D020 | CALL PROC | Вызов пдпрограммы |
| 19 | 0130 | ADD R3 R0 | + |
| 1A | A401 | SB R4 01 | -1 |
| 1B | 041D | JZ M2 | Проверка на 0 |
| 1C | 0314 | JMP M1 | Переход |
| 1D | C304 | MV R3 SUM (M2) | Записать в сумму |
| 1E | 0000 | HLT | Остановка |
| 1F | 0000 |  |  |
| 20 | 9200 | AD R2 00 (PROC) |  |
| 21 | 0427 | JZ M3 | Проверка |
| 22 | 0131 | ADD R3 R1 (M4) | + |
| 23 | 0427 | JZ M3 | Проверка |
| 24 | A201 | SB R2 01 | -1 |
| 25 | 0427 | JZ M3 | Проверка |
| 26 | 0322 | JMP M4 | Переход |
| 27 | 0500 | RET (M3) | Возвращение |
|  |  |  |  |

4. Разработка алгоритма и структуры программы

4.1. Алгоритм работы ЭВМ



5. Микропрограммная реализация ЭВМ

5.1. Распределение регистров

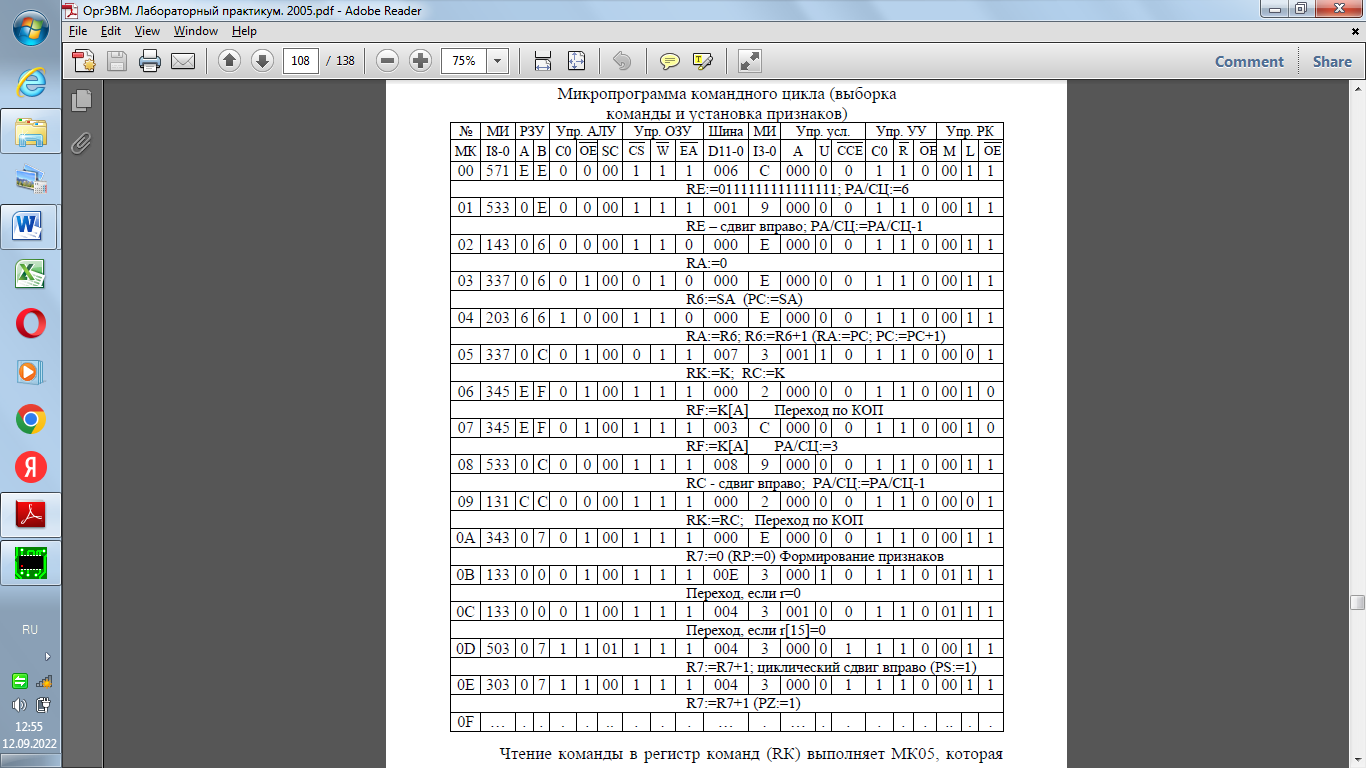
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | R0-R7 |  |  | R8-R15 |
| 0 | r0 |  | 8 | r0 (SP) |
| 1 | r1 |  | 9 |  |
| 2 | r2 |  | 10 |  |
| 3 | r3 |  | 11 |  |
| 4 | r4 |  | 12 |  |
| 5 | r5 |  | 13 | Буферный регистр команд |
| 6 | r6 (PC) |  | 14 | Регистр константы |
| 7 | r7 (RP) |  | 15 | Счетчик адреса ЗУ |

5.2. Дешифрация кода

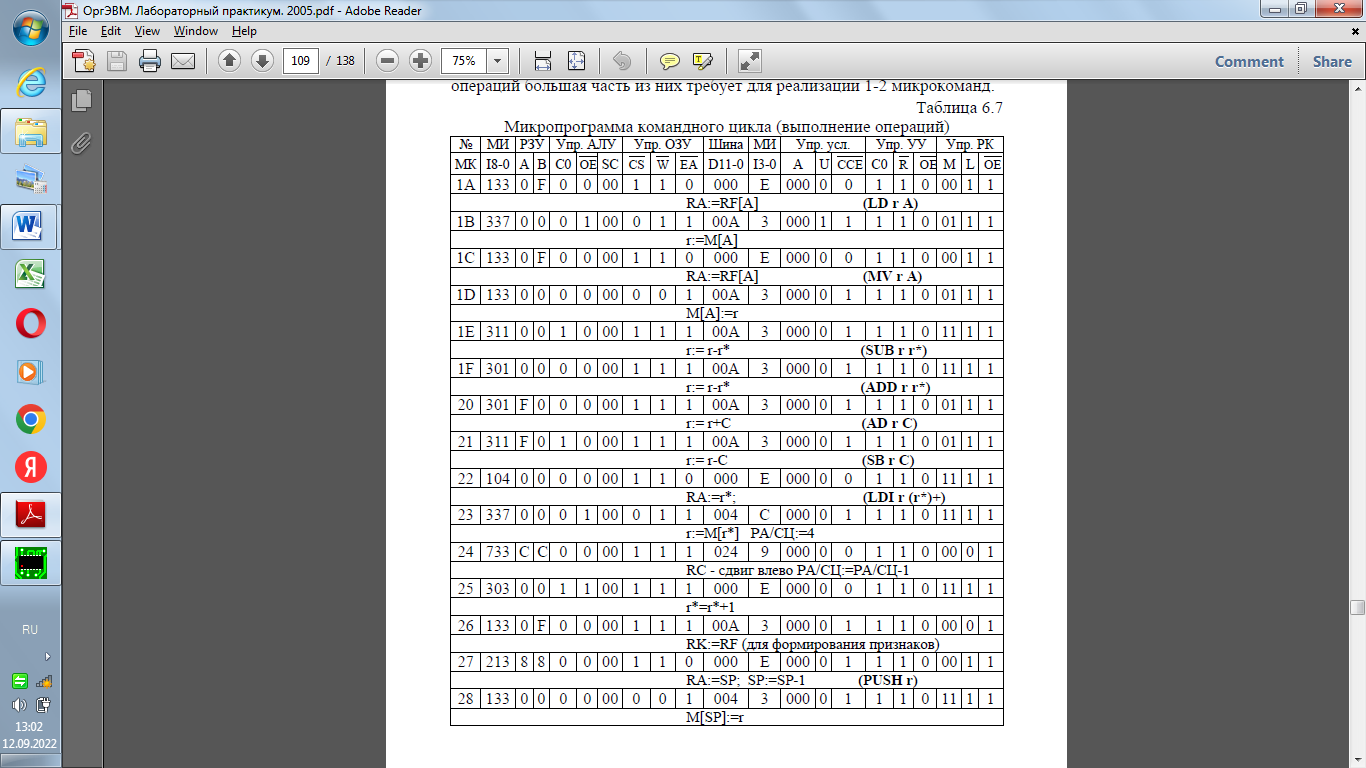
|  |  |  |
| --- | --- | --- |
| Мнемоника | Код операции | Адрес первой МК |
| ADD r r\* | 01 | 1E |
| SUB r r\* | 02 | 1F |
| AD r C | 9 | 20 |
| SB r C | A | 21 |
| LD r A | B | 1A |
| MV r A | C | 1C |
| LDI r r\* | 0E | 2F |
| PUSH r (SP) | 07 | 22 |
| POP r (SP) | 06 | 24 |
| JMP A | 03 | 27 |
| JZ A | 04 | 26 |
| CALL (SP) A | D | 28 |
| RET (SP) | 05 | 2B |
| HLT A | 00 | 2D |

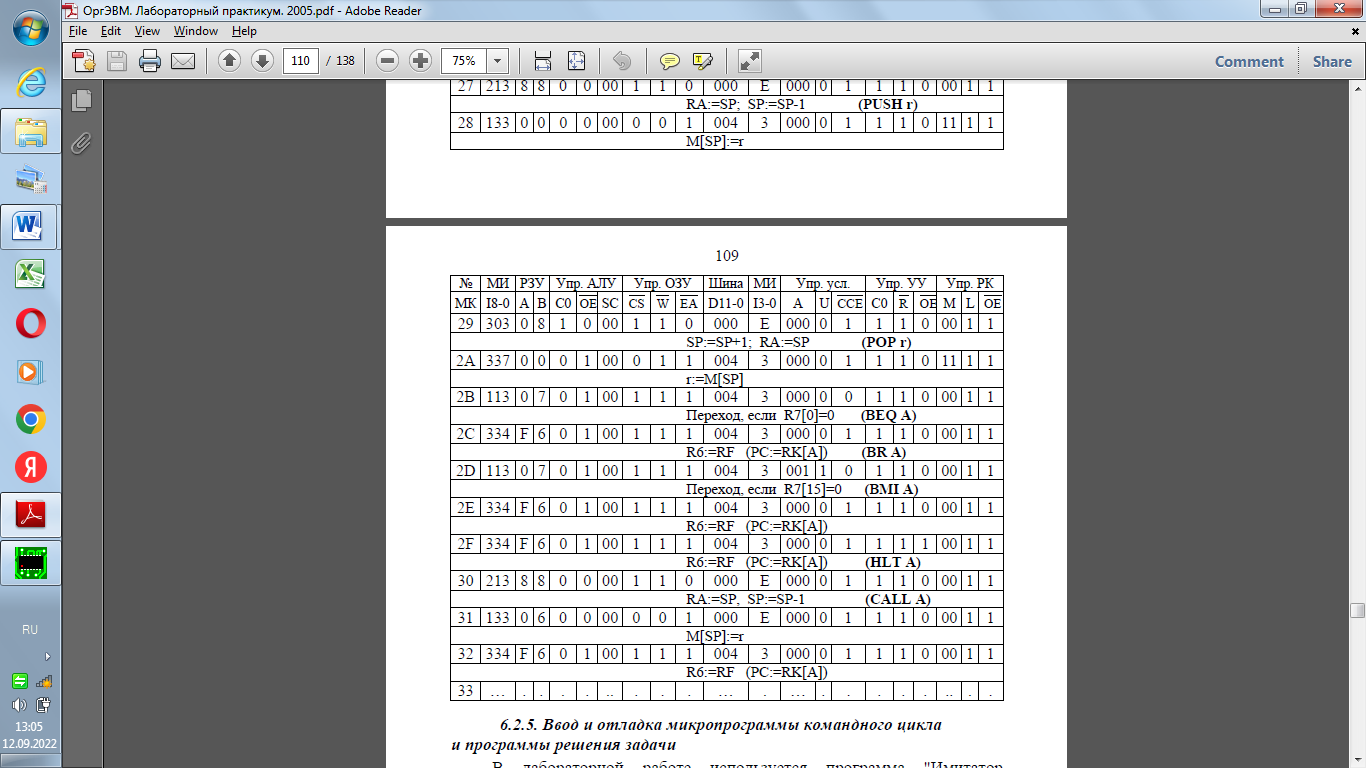
5.3. Микропрограммная реализация

5.3.1 Микропрограмма командного цикла (выборка команды и установка признаков)



5.3.2. Микропрограмма командного цикла (выполнения операций)





6. Вывод

В ходе выполнения лабораторной работы была разработана и изучена учебная ЭВМ; разработана и реализована система команд, написана программа решения задачи, которая помещена в ОЗУ. По сравнению с предыдущей лабораторной работой, система команд расширена. В дополнение к прямой были добавлены следующие виды адресации: регистровая, неявная, регистровая преинкрементная и постдекрементная, непосредственная. Введение различных видов адресации усложнило командный цикл, но сделало написание программы удобнее и понятнее для программиста.